R E M A R K S

Careful review and examination of the subject application are noted and appreciated.

SUPPORT FOR THE CLAIM AMENDMENTS

Support for the claim amendments can be found in the specification, for example, on page 6, line 1-5, page 32 line 18 thru page 33 line 8, page 34 lines 11-18, page 40 line 7-10, FIGS. 6-7 and claims 12, 13, 17 and 18 as originally filed. Thus, no new matter has been added.

CLAIM REJECTIONS UNDER 35 U.S.C. §103

The rejection of claims 1-8 and 10 under 35 U.S.C. §103(a) as being unpatentable over MacCrisken '348 in view of Keller et al. '709 (hereafter Keller) is respectfully traversed and should be withdrawn.

The rejection of claim 9 under 35 U.S.C. §103(a) as being unpatentable over MacCrisken '348 in view of Keller and Ko '892 is respectfully traversed and should be withdrawn.

MacCrisken concerns an adaptive data compression system (Title). Keller concerns a run-time routing for programmable logic devices (Title). Ko concerns an EFM/DVD demodulator (Title). In contrast, the present invention provides a method of generating a file suitable for programming a programmable logic device. The

method generally comprises the steps of (A) generating a programming item from a plurality of parameters that define a program for the programmable logic device, (B) compressing the programming item to present a compressed item, (C) storing the programming item in a programming field of the file in response to generating and (D) storing the compressed item in a non-programming field of the file in response to compressing.

Claim 1 provides a step for storing a programming item in a programming field of a file. Despite the assertion on page 4 of the Office Action, the text in column 4, lines 15-17 of Keller appear to be silent regarding storing data **in a programming field** of a file. Therefore, MacCrisken and Keller, alone or in combination, do not appear to teach or suggest a step for storing a programming item in a programming field of a file as presently claimed.

Claim 1 further provides storing a compressed item in a non-programming field of a file. Despite the assertion on page 3 of the Office Action, the text in column 4, lines 62-67 of MacCrisken appear to be silent regarding storing compressed data **in a non-programming field** of a file. Therefore, MacCrisken and Keller, alone or in combination, do not appear to teach or suggest a step for storing a compressed item in a non-programming field of a file as presently claimed.

Furthermore, *prima facie* obviousness has not been established for lack of clear and particular motivation to combine the references. In particular, the assertion on page 4 of the Office Action that "these bits level could be written in many different programming languages" does not appear to be based on either reference or knowledge generally available to one of ordinary skill in the art (MPEP §2142). Therefore, the asserted motivation appears to be merely a conclusory statement.

Furthermore, the references appear to be non-analogous art. MacCrisken has a primary U.S. classification of 375/122. In contrast, Keller has a primary U.S. classification of 716/14. However, no evidence has been provided in the Office Action that MacCrisken is either (i) within the Applicants' field of endeavor or (ii) reasonably pertinent to the particular problem with which the Applicants' were concerned (MPEP §2141.01(a)). Due to a lack of evidence to the contrary, the U.S. Patent and Trademark Office classifications appear to show that the references are non-analogous art and thus the proposed combination is not obvious. As such, the claimed invention is fully patentable over the cited references and the rejection should be withdrawn.

Claim 2 provides a step for storing at least one of a plurality of parameters in a second non-programming field of a file. Despite the assertion on page 4 of the Office Action, the text in column 5, lines 15-17 of Keller appear to be silent

regarding non-programming fields of files. The text of Keller cited in the Office Action reads:

> Thus, usage of the term "bitstream" is intended to encompass sequences of programming bits for both partial and full reconfiguration.

Nowhere in the above text, or in any other section does Keller appear to discuss non-programming fields of a file. Therefore, MacCrisken and Keller, alone or in combination, do not appear to teach or suggest a step for storing at least one of a plurality of parameters in a second non-programming field of a file as presently claimed. As such, claim 2 is fully patentable over the cited references and the rejection should be withdrawn.

Claim 4 provides a dictionary generated independently of a compressing step. Despite the assertion on page 4 of the Office Action, the text in column 6, lines 57-68 of MacCrisken appear to be silent regarding how a dictionary is generated. The text of MacCrisken cited in the Office Action reads:

> Each character of raw data is encoded using the bigram table for the previous input character. Thus, for the input string "the quick", the letter "h" is encoded using the bigram table for letter "t". To find the "t" bigram table, the binary value of the raw character "t" is used as a pointer to look up the bicode for "t" in the ABtrans table 64. If "t" is one of the BxMax (typically sixty) most common characters in the data sample used to build the E Table 60, it will have a nonzero bicode.
>
> The bicode for "t" is then used as a pointer to look up in the Bindex table 66 the address of the "t" bigram table 62.

Nowhere in the above text, or in any other section, does MacCrisken appear to indicate that the bigram table is **generated independently**

of compressing data. Therefore, MacCrisken and Keller, alone or in combination, do not appear to teach or suggest a dictionary generated independently of a compressing step as presently claimed. As such, claim 4 is fully patentable over the cited references and the rejection should be withdrawn.

Claim 6 provides a step for encoding a compressed item from a binary representation to a symbol representation. In contrast, MacCrisken appears to be silent regarding mapping information from a binary representation to a symbol representation. In particular, column 6, lines 5-7 of MacCrisken contemplates that the term "encode" is used interchangeably with the term "compress". Thus, the "encoding" used in column 5, lines 60-67 of the MacCrisken (cited on page 5 of the Office Action) appears to be speaking about compressing input data, not transforming data **already compressed** from a binary representation into a symbol representation. Therefore, MacCrisken and Keller, alone or in combination, do not appear to teach or suggest a step of encoding a compressed item from a binary representation to a symbol representation as presently claimed. As such, claim 6 is fully patentable over the cited references and the rejection should be withdrawn.

Claim 7 provides a step for encoding a compressed item (from claim 6) and a step for mapping a symbol representation to a character representation in response to the encoding. In contrast,

the text in column 1, lines 23-28 of MacCrisken (cited on page 5 of the Office Action), appears to contemplate mapping English letters, numbers and punctuation marks into an ASCII binary codes before compressing the information. The rest of MacCrisken appears to be silent regarding mapping symbol representations into character representations **in response to encoding the compressed items**. Therefore, MacCrisken and Keller, alone or in combination, do not appear to teach of suggest a step of encoding a compressed item and mapping a symbol representation to a character representation in response to encoding as presently claimed. As such, claim 7 is fully patentable over the cited references and the rejection should be withdrawn.

Claim 8 provides a step for storing an error detection item in a second non-programming field of a file. Despite the assertion on page 5 of the Office Action, the text in column 20, lines 42-46 of MacCrisken appears to be silent regarding storing information in a non-programming field of a file. The text of MacCrisken quoted by the Office Action reads:

> Next, an error detection code is put on the end of the packet, and a byte counter, which specifies the number of bytes in the packet, is added to the front of the packet. Note that the error detection code is a standard two-byte CRC-16 code in the preferred embodiment.

Nowhere in the above text, or in any other section does MacCrisken appear to indicate that the CRC-16 code is stored **in a non-programming field of a file**. Therefore, MacCrisken and Keller,

alone or in combination, do not appear to teach or suggest a step for storing an error detection item in a second non-programming field of a file as presently claimed. As such, claim 8 is fully patentable over the cited references and the rejection should be withdrawn.

Claim 9 provides a step for validating a backup programming item with an error detection item. In contrast, the text in column 7, lines 61-63 of MacCrisken (cited on page 6 of the Office Action) states:

> When a complete packet is received, its error detection code is checked to make sure that the data received is error free.

Nowhere in the above text, or in any other section does MacCrisken discuss validating a backup programming item. In particular, the cited text of MacCrisken only appears to discuss validating individual packets each containing portions of the overall data. Therefore, MacCrisken, Keller and Ko, alone or in combination, do not appear to teach or suggest a step for validating a backup programming item with an error detection item as presently claimed.

Claim 9 further provides that the backup programming item is decompressed from a compressed item. In contrast, the validation of packets as stated in column 7, lines 61-63 of MacCrisken, appears to take place **before decompression** of the data stored within the packets. The rest of MacCrisken appears to be silent regarding a validation taking pace after decompression. Therefore, MacCrisken, Keller and Ko, alone or in combination, do

not appear to teach or suggest a step for validating a backup programming item with an error detection item as presently claimed. As such, claim 9 is fully patentable over the cited reference and the rejection should be withdrawn.

Furthermore, *prima facie* obviousness has not be established for lack of clear and particular evidence of motivation to combine MacCrisken and Keller with Ko. In particular, the assertion of page 6 of the Office Action that motivation is provided because "the extracted data items are used in synchronizing the programmable logic device data output" does not appear to be based on the teachings of MacCrisken, Keller, Ko or knowledge generally available to one of ordinary skill in the art (MPEP §2142). As such, the asserted motivation appears to be merely a conclusory statement.

Furthermore, references each appear to be non-analogous art. MacCrisken has a primary U.S. classification of 375/122. Keller, has a primary U.S. classification of 716/14. Ko has a primary U.S. classification of 341/106. However, no evidence has been provided in the Office Action that MacCrisken or Ko are either (i) within the Applicants' field of endeavor or (ii) reasonably pertinent to the particular problem with which the Applicants' were concerned (MPEP §2141.01(a)). Due to a lack of evidence to the contrary, the U.S. Patent and Trademark Office classifications appear to show that the references are non-analogous art and thus

the proposed combination is not obvious. As such, claim 9 is fully patentable over the cited references and the rejection should be withdrawn.

Accordingly, the present application is in condition for allowance. Early and favorable action by the Examiner is respectfully solicited.

The Examiner is respectfully invited to call the Applicants' representative should it be deemed beneficial to further advance prosecution of the application.

If any additional fees are due, please charge our office Account No. 50-0541.

Respectfully submitted,

CHRISTOPHER P. MAIORANA, P.C.

Christopher P. Maiorana
Registration No. 42,829
24025 Greater Mack, Suite 200
St. Clair Shores, MI 48080
(586) 498-0670

Dated: February 10, 2004

Docket No.: 0325.00488